

Name : Afsha Sayeed Khan

Concepts of Operating System

Assignment 2

Part A What will the following commands do?

- echo "Hello, World!"

It will print Hello , World

- name="Productive"

Initializing variable

- touch file.txt

Create the file having name file.txt in current directory

- ls -a

List the all file with hidden files

- rm file.txt

This command will remove the file from current directory

- cp file1.txt file2.txt

It **copies** the contents of file1.txt into file2.txt. If file2.txt does not exist → it will be created with the same content as file1.txt. If file2.txt already exists → its content will be **overwritten** with the content of file1.txt (no warning by default).

- mv file.txt /path/to/directory/

Moves file.txt into the given directory (/path/to/directory/).

- chmod 755 script.sh

Changes the **permissions** of script.sh.

Owner : read (r), write (w), execute (x) → (7)

Group : read (r), execute (x) → (5)

Others : read (r), execute (x) → (5)

- grep "pattern" file.txt

It will find or search the word pattern from the content of file

- kill PID

By default, kill sends the **SIGTERM (signal 15)** this asks the process to terminate **gracefully**.

- mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt

mkdir mydir : make a new directory called mydir.

cd mydir : go inside it.

touch file.txt : create an empty file file.txt.

echo "Hello, World!" > file.txt → put text "Hello, World!" into the file (overwrite).

cat file.txt : display the file content.

- `ls -l | grep ".txt"`

list files with details and `grep ".txt"` filter only .txt files.

- `cat file1.txt file2.txt | sort | uniq`

arrange lines alphabetically and `uniq` remove duplicate lines.

- `ls -l | grep "^d"`

show only directories (because lines starting with d mean directory).

- `grep -r "pattern" /path/to/directory/`

Search for "pattern" inside all files under a directory.

- `cat file1.txt file2.txt | sort | uniq -d`

same as before but `-d` = show **only duplicate lines** between files.

- `chmod 644 file.txt`

File permissions become:

- Owner: read & write
- Group: read only
- Others: read only

- `cp -r source_directory destination_directory`

Copy a whole directory with all files and subfolders.

- `find /path/to/search -name "*.txt"`

Find all .txt files inside given path (and subfolders).

- `chmod u+x file.txt`

Adds **execute permission** for the file's **owner (u)**. Useful for making scripts runnable.

- `echo $PATH`

prints your **PATH environment variable**.

PATH = list of directories where system looks for commands.

Part B

Identify True or False:

1. `ls` is used to list files and directories in a directory.

Answer : True

2. `mv` is used to move files and directories.

Answer : True

3. `cd` is used to copy files and directories.

Answer : False

4. `pwd` stands for "print working directory" and displays the current directory.

Answer : True

4. `grep` is used to search for patterns in files.

Answer : True

5. `chmod 755 file.txt` gives read, write, and execute permissions to the owner, and read and execute permissions to group and others.

Answer : True

7. `mkdir -p directory1/directory2` creates nested directories, creating `directory2` inside `directory1` if `directory1` does not exist.

Answer : True

8. `rm -rf file.txt` deletes a file forcefully without confirmation

Answer : True

Identify the Incorrect Commands:

1. `chmodx` is used to change file permissions.

Incorrect

2. `cpy` is used to copy files and directories.

Incorrect

3. `mkfile` is used to create a new file.

Incorrect

4. `catx` is used to concatenate files.

Incorrect

6. `rn` is used to rename files.

Incorrect

Part C

Question 1: Write a shell script that prints "Hello, World!" to the terminal.

```
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment$ mkdir os
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment$ cd os
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ vi s1.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ cat s2.sh
cat: s2.sh: No such file or directory
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ cat s1.sh
#!/bin/bash
echo "Hello, World!"

afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ chmod +x s1.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ ./s1.sh
Hello, World!
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$
```

Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable. Question

```
afsha@LAPTOP-DBBDT8K x + v
#!/bin/bash
name="CDAC Mumbai"
echo "variable name is : $name"

~
~
~
~

afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ vi s3.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ chmod +x s3.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ ./s3.sh
variable name is : CDAC Mumbai
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$
```

3: Write a shell script that takes a number as input from the user and prints it. Question

```
afsha@LAPTOP-DBBDT8K x + v
#!/bin/bash
echo "Enter the number :"
read Number
echo "The number is : $Number"
~
~
~
~

afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ vi s2.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ chmod +x s2.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ ./s2.sh
Enter the number :
10
The number is : 10
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$
```

4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.

```
afsha@LAPTOP-DBBDT8K x + v
#!/bin/bash
echo "Enter the first number"
read x
echo "Enter the second number"
read y
(( sum=x+y ))
echo "The result of sum is :$sum"
~
~
~
~

afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ vi s3.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ vi addition.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ chmod +x addition.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ ./addition.sh
Enter the first number
5
Enter the second number
6
The result of sum is :11
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ ./addition.sh
Enter the first number
5
Enter the second number
3
The result of sum is :8
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$
```

Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".

```
afsha@LAPTOP-DBBDT8K x + v
#!/bin/bash
echo -n "Enter the number : "
read num
if [[ ($num -lt 10) && ($num%2 -eq 0) ]]; then
    echo "Even number"
else
    echo "Odd number"
fi
??
??
??

./evenodd.sh: Line 8: Syntax error: unexpected end of file
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ vi evenodd.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ chmod +x evenodd.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ ./evenodd.sh
Enter the number : 7
Odd number
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$
```

Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.

```
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ vi forloop.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ chmod +x forloop.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ ./forloop.sh
Number is : 1
Number is : 2
Number is : 3
Number is : 4
Number is : 5
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ cat forloop.sh
#!/bin/bash
for i in {1..5}
do
    echo "Number is : $i"
done
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ vi forloop.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ ./forloop.sh
1
2
3
4
5
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$
```

Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.


```
afsha@LAPTOP-DBBDT8K x + v
#!/bin/bash
i=1
while [ $i -le 5 ]
do
    echo "Number : $i"
    ((i++))
done
~
~
~
~
~
~
~
~
~
~
"whileloop.sh" 7L, 72B 7,4

Odd number
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ vi whileloop.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ chmod +x whileloop.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ ./whileloop.sh
Number : 1
Number : 2
Number : 3
Number : 4
Number : 5
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$
```

Question 8: Write a shell script that checks if a file named "file.txt" exists in the current directory. If it does, print "File exists", otherwise, print "File does not exist".

```
afsha@LAPTOP-DBBDT8K x + v
#!/bin/bash
if [[ -f "file.txt" ]]; then
    echo "File exists"
else
    echo "File does not exist"
fi
~
~
~
~
~
```

```
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ vi file.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ chmod file.sh
chmod: missing operand after 'file.sh'
Try 'chmod --help' for more information.
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ chmod +x file.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ ./ file.sh
-bash: ./: Is a directory
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ ./file.sh
File does not exist
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$
```

Question 9: Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.

```
afsha@LAPTOP-DBBDT8K x + v
#!/bin/bash
echo "Enter the number"
read num
if [[ (num -gt 10) ]]; then
    echo "Number is greater then : $num"
else
    echo "Number is less then : $num"
fi
~
~
~
~
```

```
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ vi greater.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ chmod +x greater.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ ./greater.sh
Enter the number
8
Number is less then : 8
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ ./greater.sh
Enter the number
15
Number is greater then : 15
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$
```

Question 10: Write a shell script that uses nested for loops to print a multiplication table for numbers from 1 to 5. The output should be formatted nicely, with each row representing a number and each column representing the multiplication result for that number.

```
afsha@LAPTOP-DBBDT8K x + v
#!/bin/bash
echo "Multiplication Table (1 to 5)"
echo "-----"

for i in {1..5}
do
    for j in {1..5}
    do
        # Print the product with tab spacing
        printf "%4d" $((i * j))
    done
    echo # Move to next line after each row
done

afsha@LAPTOP-DBBDT8K:~/LinuxAssignment/os$ vi file.sh
afsha@LAPTOP-DBBDT8K:~/LinuxAssignment/os$ vi multiplication.sh
afsha@LAPTOP-DBBDT8K:~/LinuxAssignment/os$ chmod +x multiplication.sh
afsha@LAPTOP-DBBDT8K:~/LinuxAssignment/os$ ./multiplication.sh
Multiplication Table (1 to 5)
-----
 1  2  3  4  5
 2  4  6  8 10
 3  6  9 12 15
 4  8 12 16 20
 5 10 15 20 25
afsha@LAPTOP-DBBDT8K:~/LinuxAssignment/os$
```

Question 11: Write a shell script that uses a while loop to read numbers from the user until the user enters a negative number. For each positive number entered, print its square. Use the break statement to exit the loop when a negative number is entered.

```
afsha@LAPTOP-DBBDT8K × + ▾
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ vi w.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ chmod +x w.sh
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ ./w.sh
Enter the no to find square root of number
2
square root is 4
Enter the no to find square root of number
3
square root is 9
Enter the no to find square root of number
4
square root is 16
Enter the no to find square root of number
5
square root is 25
Enter the no to find square root of number
0
square root is 0
Enter the no to find square root of number
-2
Exiting negative no entered
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$ cat w.sh
#!/bin/bash
while true
do
    echo "Enter the no to find square root of number"
    read num
    if [[ $num -lt 0 ]]; then
        echo "Exiting negative no entered"
        break
    else
        ((square=num*num))
        echo "square root is $square"
    fi
done
afsha@LAPTOP-DBBDT8KE:~/LinuxAssignment/os$
```